LINEARIZATION ALGORITHMS AND HEURISTICS FOR COMPUTER-AIDED CONTROL ENGINEERING

James H. Taylor
Alfred J. Antoniotti
Control Systems Laboratory
General Electric Corporate Research and Development
PO Box 8, Schenectady, New York 12301

ABSTRACT

Generating a linearized dynamic system model corresponding to a nonlinear system at a specific operating point provides a critical bridge between nonlinear simulation and linear analysis and design. Obtaining such a linearized model by numerical means (taking finite differences) is by no means a simple task. In some cases obtaining an accurate estimate of the derivative of a nonlinear function requires careful selection of the perturbation used in taking finite differences; in other cases the derivative is not defined and simple numerical differentiation may lead to totally meaningless results. In this paper we present algorithms and heuristic logic for the accurate and robust linearization of nonlinear dynamic system models.

This research [1] in robust linearization methods has yielded new conventional methods and algorithms for linearization, as well as a new expert system to aid the controls engineer in determining linearized models for nonlinear systems. Some important aspects of this work include: an approach to minimize the effects of truncation and round-off errors incurred through numerical differentiation, and techniques for accurately identifying certain discontinuities in the mathematical description of a nonlinear systems and other problems that make linearization difficult or meaningless. We focus here on conventional methods and algorithms, which incorporate all of the "expert knowledge" gained in the course of this effort.

1. INTRODUCTION AND OVERVIEW

A substantial research effort at GE CR&D has been fucussed on the development of general environments for Computer-Aided Control Engineering (CACE), covering the traditional span from nonlinear modeling and simulation of the system to be controlled, through linear analysis and design, and culminating in nonlinear simulation of the controlled system [1]. This included both conventional software development [2,3] and the investigation of expert system applications in a package referred to as CACE-III [4,5,6,7]. Throughout this work, linearization has played a pivotal role in the functionality of the environment [1,6], and much has been learned. The results presented here represent the algorithmic and heuristic knowledge gained during this research.

The software package used for the development of linearization techniques has been SIMNON [2], which supports modeling and simulation of nonlinear systems and has been extended with routines for equilibrium finding and linearization [3]. This package was incorporated both in the Federated System [2] and in CACE-III. The history of our studies of linearization demonstrate how conventional and expert-system software development can be synergistic – the first algorithms were inplemented conventionally in SIMNON+, then "expert aided" in CACE-III, then improved in SIMNON+ to exploit the knowledge gained in expert system development without the overhead of an ancillary expert system shell. Finally, SIMNON+ has been made an integral part of MEAD [9] and was again refined in the process.

The purpose of all this was to develop the most appropriate linearized model for a nonlinear system at a given operating point, and to qualify that model in general terms (e.g., to establish what types of nonlinearities exist and provide a measure of validity of the linear model). Eventually, a more versatile package may be developed to include the use of describing function techniques to characterize certain system nonlinearities and to form a base from which further, more extensive treatment can proceed (e.g., describing function synthesis methods, [10,11]), as outlined by Taylor and Frederick [6].

2. SIMNON+ LINEARIZATION ALGORITHM

We begin with a formal definition of the linearization problem and an overview of some of the principles involved in numerical differentiation [1]. Details of SIMNON+'s linearization algorithm follow, including numerical properties and heuristic logic. This discussion is strictly limited to the conventional implementation of linearization, although it is based on lessons learned in developing an expert-system rule base for extracting linearized models.

[1]Some of this work was performed by Mr. Alfred J. Antoniotti as part of a Master's project for the Rensselaer Polytechnic Institute, under the guidance of Dr. James H. Taylor and while employed by GE CR&D.

[2]SIMNON [8] was developed by the Department of Automatic Control at Lund Institute of Technology, Lund, Sweden, and is a commercial product of SSPA Systems, Goteborg, Sweden; versions extended by GE CR&D are called SIMNON+.

2.1. Problem Statement

SIMNON+ addresses the following linearization problem: Consider a system in the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{1}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}) \tag{2}$$

where $\mathbf{x}$ represents the state vector of dimension n, $\mathbf{u}$ denotes the input vector of dimension m, and $\mathbf{y}$ is the output vector of dimension p. Both functions $\mathbf{f}$ and $\mathbf{h}$ are nonlinear in general.

Finding the system equilibrium corresponding to a given constant input value $\mathbf{u_0}$ is usually the first step:

$$\mathbf{u_0} \rightarrow \mathbf{x_0} : \mathbf{f}(\mathbf{x_0}, \mathbf{u_0}) = 0 \tag{3}$$

Then the output at this equilibrium is given by:

$$\mathbf{y_0} = \mathbf{h}(\mathbf{x_0}, \mathbf{u_0}) \tag{4}$$

The linearized model about the operating point $(\mathbf{x_0}, \mathbf{u_0})$ is defined as follows:

$$\delta\dot{\mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\delta\mathbf{u} \tag{5}$$

$$\delta\mathbf{y} = \mathbf{C}\delta\mathbf{x} + \mathbf{D}\delta\mathbf{u} \tag{6}$$

where $\delta\mathbf{x} = \mathbf{x} - \mathbf{x_0}$, $\delta\mathbf{u} = \mathbf{u} - \mathbf{u_0}$, and $\delta\mathbf{y} = \mathbf{y} - \mathbf{y_0}$. The matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ are defined by:

$$\mathbf{A} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{\mathbf{x_0}, \mathbf{u_0}} \quad , \quad \mathbf{B} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right]_{\mathbf{x_0}, \mathbf{u_0}} \quad , \tag{7}$$

$$\mathbf{C} = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]_{\mathbf{x_0}, \mathbf{u_0}} \quad , \quad \mathbf{D} = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right]_{\mathbf{x_0}, \mathbf{u_0}} \tag{8}$$

This linearized model is valid for limited variations in the states and inputs about the equilibrium, provided that the above partials exist. The appropriate range of variation of the states and inputs depends on how nonlinear the system is at $(\mathbf{x_0}, \mathbf{u_0})$. This process is sometimes called ***small-signal linearization.***

2.2 Linearization by Numerical Differentiation

The derivative of a function $f(x)$ at a point x_0 is defined by

$$\frac{df}{dx} = \lim_{\delta \to 0} \frac{f(x_0 + \delta) - f(x_0 - \delta)}{2\delta} \tag{9}$$

provided that the derivative at x_0 exists. A sufficient condition ensuring the validity of Eqn. 9 is that $f(x)$ should have a Taylor-series expansion about the point x_0. (Note that the notation here and below is simplified by confining the discussion to a scalar function of one variable – the extension to the general case, Eqns. 1 and 2, is obvious. Also, the symbols x and δ are used quite freely throughout to signify a value and perturbation in either a state or an input.)

One way to compute an estimate of the derivative of a function is to calculate a finite central difference:

$$Df^{(\delta)} = \frac{f(x_0 + \delta) - f(x_0 - \delta)}{2\delta} \tag{10}$$

Complete accuracy would require the perturbation to be infinitesimal and the number of significant figures in the function evaluations to be infinite.

From a theoretical standpoint, any error incurred in using Eqn. 10 is an indication of the curvature of the nonlinearity. If the nonlinear function can be represented by a Taylor-series expansion, then it is easily seen that the accuracy of Eqn. 10 depends (in part) on how dominant the constant, linear, and quadratic terms are in relation to higher-order terms in the expansion: the central-difference calculation yields an exact result only for quadratic functions (see Appendix A). Error arises through neglecting the higher-order nonlinear terms of $f(x)$, or truncation of its Taylor-series expansion. Such error is thus called ***truncation error.*** Since truncation error increases with perturbation size, the obvious remedy is to use the smallest perturbation possible.

Another source of error devolves from the fact that computers have limited precision. Specifying an arbitrarily small perturbation generally does not work, as the quantities $f(x+\delta)$ and $f(x-\delta)$ become nearly equal. When this happens the central difference loses accuracy; results are said to be dominated by ***round-off error.*** It is difficult to say what perturbation size introduces round-off error, since this depends on the particular problem under analysis. However, the deleterious effects of round-off error can be greatly reduced by performing calculations with higher precision. Doing computations in double precision as opposed to single precision can have a dramatic impact on the significance of round-off error. Calculations are performed with single precision in SIMNON, however, so round-off error is a valid concern.

Thus the tradeoff between truncation and round-off effects must be considered in specifying the proper perturbation size for linearization. The curvature of the nonlinear function to be linearized plus the magnitude (or units) of the variables and of the function near the point of linearization together with machine precision define the problem.

In most applications minimum total error is obtained over a rather wide range of perturbation sizes, rather than by a unique value. A typical plot of linearization error versus perturbation size would thus have the form of a "valley" region of minimal error for some range of δ; on either side error rises, as illustrated in Fig. 1. The truncation-error region is generally rather smooth, with an initial trend that

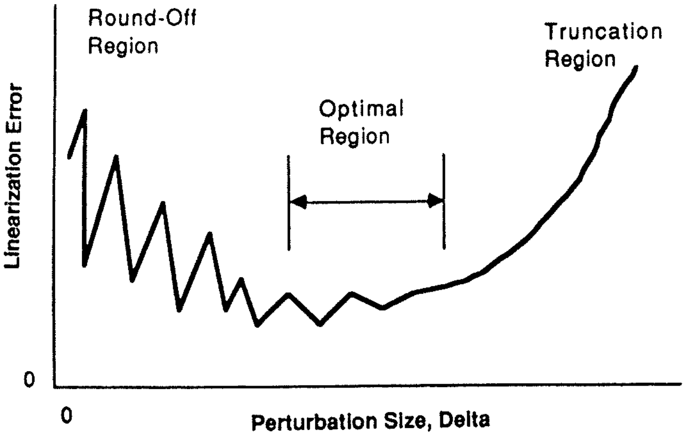

Figure 1: Conceptual Relation Between Linearization Error and Perturbation Size

is dominated by the first neglected term in the Taylor series expansion at x_0; for larger values of δ the higher-order terms begin to show their influence. The jagged nature of the curve in the round-off region is an indication of the "random" behavior in the computation of the central difference. For this range of δ, the value of $f(x+\delta) - f(x-\delta)$ seems to vary erratically; however, there is an underlying error growth that is inversely proportional to δ (due to the denominator of Eqn. 10). For calculations performed with double precision, the region of minimal error may be many orders of magnitude (in perturbation size) broader, and the error would generally be much less over this range; nevertheless, the same valley-shaped curve governs the relation between linearization error and δ. Finally, observe that an optimal δ must be found *for each partial derivative*, i.e., for each element in the system matrices **A**, **B**, **C**, and **D**; therefore, it is necessary to determine an optimum *matrix* $[\delta_{i,j}]$ corresponding to each system matrix.

2.3. SIMNON+ Linearization

SIMNON+ does not simply compute a single central difference estimate (Eqn. 10) with some fixed δ to determine an approximation to the desired derivative. In general terms, it calculates one estimate based on perturbation δ, denoted by $Df^{(\delta)}$, and another estimate based on perturbation 2δ, denoted by $Df^{(2\delta)}$, compares these values for error control to determine an optimal value of δ, and finally returns a weighted combination of the central difference estimates [3] for that δ:

$$\widehat{Df}^{(\delta)} = [4Df^{(\delta)} - Df^{(2\delta)}]/3 \tag{11}$$

[3]Note: the equations in this section are written for conceptual clarity, and are not recommended for implementation of computer algorithms. For example, care must be taken to avoid unnecessary round-off errors that would be incurred if these formulas were coded literally.

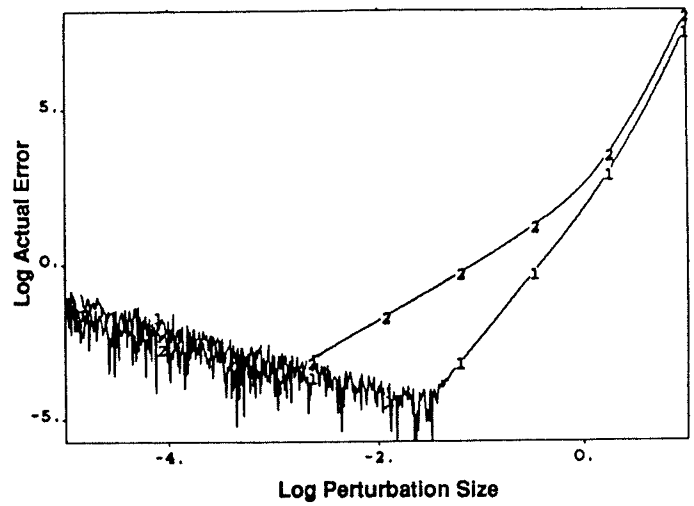

Figure 2: Linearization Error for $\widehat{Df}^{(\delta)}$ ('1') and $Df^{(\delta)}$ ('2') in x^9 Example

This calculation is an extension of Richardson extropolation ([1]; cf. Dahlqvist and Bjorck [12]). In Appendix A it is shown that Eqn. 11 eliminates truncation error in all terms up to (not including) the fifth power in any nonlinear function expressible in a Taylor series expansion. The linearization error from such a combination of derivative estimates still exhibits the general behavior shown in Fig. 1 as the perturbation δ varies, but the magnitude of the error in the optimal region is smaller. Figure 2 shows a comparison of the linearization error incurred in estimating the derivative of x^9 about $x_0 = \sqrt[9]{2}$ with $\widehat{Df}^{(\delta)}$ (curve 1) and $Df^{(\delta)}$ (curve 2) over a range of perturbation sizes. The trend in round-off is still inversely proportional to δ, but the behavior at the onset of truncation error dominance is governed by δ^5, as this is the lowest-order term in the error. Note that the curves are plotted on a log-log scale.

The perturbation used in the calculation of the final derivative estimate per Eqn. 11 is obtained via a routine that examines the tradeoff between truncation and round-off error in an effort to select a δ yielding lowest overall error. The characterization of linearization error is an important point in this analysis and procedure. It is necessary to estimate the amount of error incurred by using a particular δ, and to be able to classify such error (for example, is the error dominated by truncation or round-off).

The available measure of linearization error is based on comparing the finite-difference linearization estimates calculated using perturbations δ and 2δ. The analysis in Appendix A demonstrates that the error in Eqn. 10 at the onset of truncation is dominated by

$$e^{(\delta)} = \frac{1}{3}\left|Df^{(2\delta)} - Df^{(\delta)}\right| \tag{12}$$

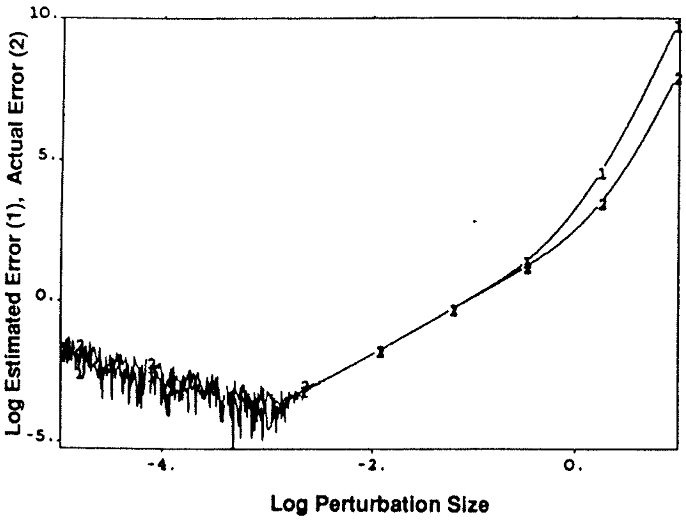

Figure 3: Error Estimate $e^{(\delta)}$ ('1') and Actual Error ('2') for $Df^{(\delta)}$, x^9 Example

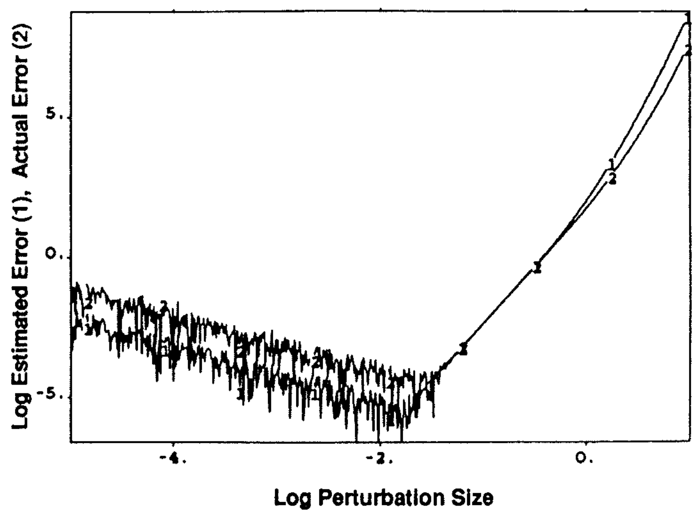

Figure 4: Error Estimate $\hat{e}^{\delta}$ ('1') and Actual Error ('2') for $\widehat{Df}^{(\delta)}$, x^9 Example

By a similar analysis, the error in Eqn. 11 at the onset of truncation is dominated by $\widehat{e}^{(\delta)}$, given by:

$$\widehat{e}^{(\delta)} = \frac{1}{15}\left|\widehat{Df}^{(2\delta)} - \widehat{Df}^{(\delta)}\right| \tag{13}$$

The above error estimates are based on the assumption that the linearization error is dominated by truncation effects. Note, however, that these estimates are themselves finite differences, and thus also subject to round-off error as δ becomes small. This effect has not been analyzed, but it is noteworthy that the estimated linearization error $e^{(\delta)}$ tracks the actual linearization error for $Df^{(\delta)}$ very closely throughout the round-off region, as well as during truncation. This is illustrated in Fig. 3, which shows these calculations for the derivative of x^9 about $x_0 = \sqrt[9]{2}$. The same observation holds for the derivative estimate $\widehat{Df}^{(\delta)}$ and error estimate $\hat{e}^{(\delta)}$, as shown in Fig. 4 which shows excellent agreement between actual and estimated error in the truncation-error region, and respectable tracking during round-off (there is an appreciable offset of about one order of magnitude in this case).

2.3.1. Bases for a Linearization Heuristic

This section begins with an overview of a heuristic approach to the linearization problem. The *ad hoc* approach is based on concepts from [6], and justified by results obtained from the analysis of several general types of nonlinear functions. Then the logic in the SIMNON+ linearization routine OPTDELTA is presented in the section that follows.

Before proceeding to find an optimal perturbation, several tests should be made to ensure the function $f(x)$ is in fact differentiable at the point x_0. Certain tests have been devised, based on a number of observations concerning the distinctively different error estimate behaviors exhibited by functions that are differentiable at x_0, discontinuous at that point, or have a discontinuous or infinite derivative at x_0. To demonstrate these behaviors, we define four examples:

Example 1 (Differentiable Function):

$$f_1 = x^9 \tag{14}$$

Example 2 (Discontinuous Function):

$$f_2 = x^9 + 0.5u(x - x_0) \tag{15}$$

where the unit step $u(x - x_0)$ is:

$$u(x - x_0) = \begin{cases} 0 & \text{if } x < x_0 \\ 1 & \text{otherwise} \end{cases} \tag{16}$$

Example 3 (Function with Infinite Derivative):

$$f_3 = x^9 + 0.5rt(x - x_0) \tag{17}$$

where the "root" function $rt(x - x_0)$ is:

$$rt(x - x_0) = \begin{cases} -\sqrt{|x - x_0|} & \text{if } x < x_0 \\ \sqrt{|x - x_0|} & \text{otherwise} \end{cases} \tag{18}$$

Example 4 (Function with Discontinuous Slope):

$$f_4 = x^9 + 0.5r(x - x_0) \tag{19}$$

where the ramp function $r(x - x_0)$ is:

$$r(x - x_0) = \begin{cases} 0 & \text{if } x < x_0 \\ x - x_0 & \text{otherwise} \end{cases} \tag{20}$$

Curves showing the variation in the error estimates $e^{(\delta)}$ (curve '1') and $\hat{e}^{(\delta)}$ (curve '2') versus the perturbation size for Examples 1 through 3 are shown

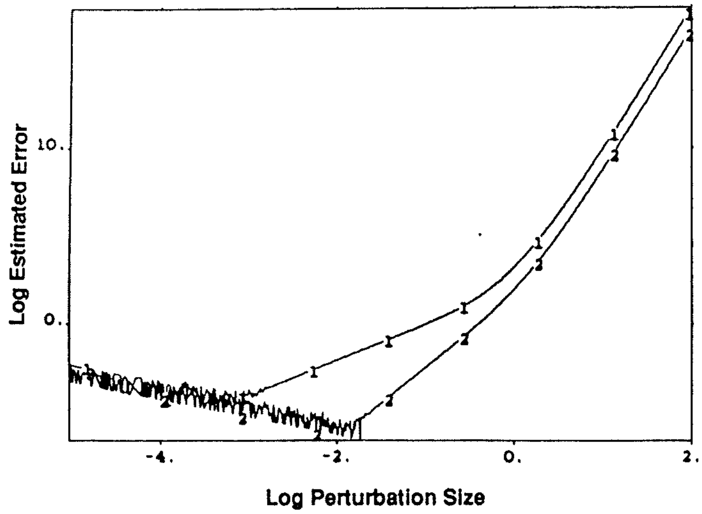

Figure 5: Error Estimates $e^{(\delta)}$ ('1') and $\hat{e}^{(\delta)}$ ('2') in Example 1

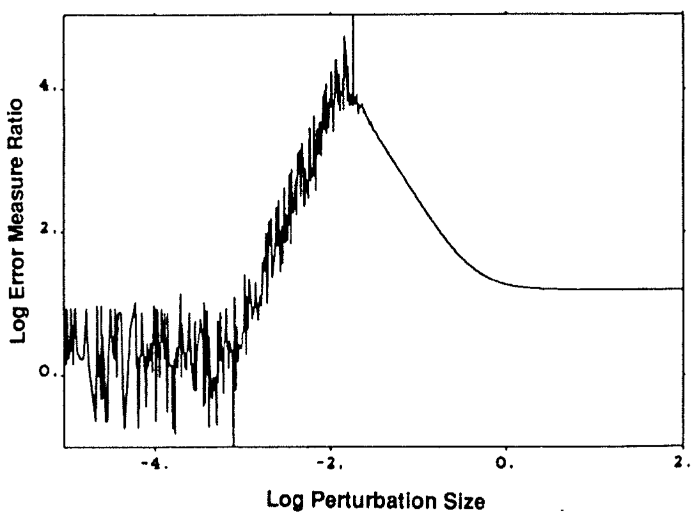

Figure 6: Ratio of the Error Estimates $e^{(\delta)}/\hat{e}^{(\delta)}$ in Example 1

in Figs. 5, 7 and 9. A study of these data reveals the importance of an indicator called the *error estimate ratio*, i.e., $e^{(\delta)}/\hat{e}^{(\delta)}$; these metrics are portrayed in Figs. 6, 8 and 10. These plots are discussed below:

1. Differentiable Functions (Example 1) - the plots in Fig. 5 depict the error estimate behavior, as discussed above. The error estimate ratio, Fig. 6, shows that this metric attains a maximum at a value of δ that is very nearly optimal with respect to the $\hat{e}^{(\delta)}$ error curve. It is also seen from these figures that all values of δ such that the error estimate ratio is greater than 100 are near or within the optimal region of δ. This behavior is typical of differentiable functions.

2. "Infinite Derivative" Functions (Examples 2, 3)

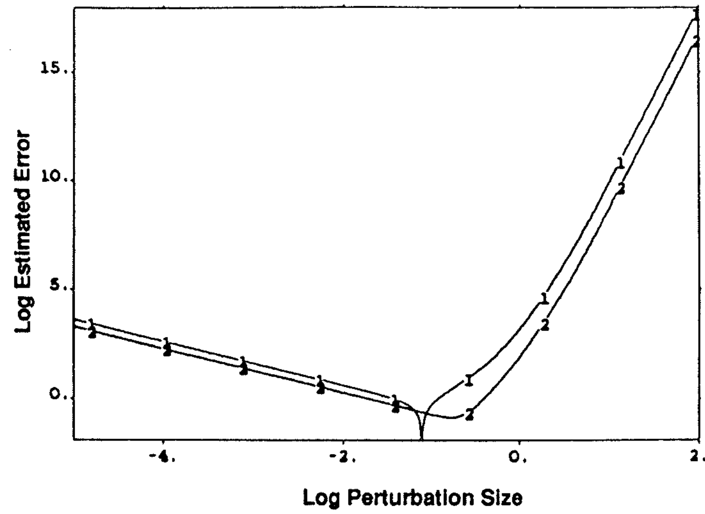

Figure 7: Error Estimates $e^{(\delta)}$ ('1') and $\hat{e}^{(\delta)}$ ('2') in Example 2

- the plots in Figs. 7 and 9 show the behavior of the error estimates for a discontinuous function and for one with a "smooth-but-infinite-derivative" characteristic, and Figs. 8 and 10 portray the corresponding linearization error estimate ratios. Note the virtual disappearance of round-off effects in the curves for this and the other undifferentiable function (Example 4). As mentioned in Section 2.2, round-off error occurs in the central difference estimate when $f(x+\delta)$ and $f(x-\delta)$ are nearly equal – this does not occur for discontinuous functions, and for functions such as the 'root' $rt(x)$ and ramp $r(x)$ this difference does not approach zero *fast enough* for round-off effects to dominate. Also note that for the "infinite derivative" functions the error estimate ratio curves are ***constant*** for small perturbation sizes. These properties readily distinguish these types of undifferentiable functions from other cases.

3. Discontinuous Functions (Example 2) - the plots in Fig. 11 show the behavior of *central difference ratios* based on derivative estimates $Df^{(\delta/2)}$, $Df^{(\delta)}$, and $Df^{(2\delta)}$ for a discontinuous function involving the step function $u(x-x_0)$. It is easily seen (make a simple sketch or refer to [6]) that, for small δ, the value of Df halves as the perturbation doubles in such cases. Thus we expect the ratio of $Df^{(2\delta)}$ to $Df^{(\delta)}$ to approach 0.5 as the perturbation size approaches zero, as shown in Fig. 11. If the function has a component with smooth-but-infinite derivative, as in Example 3, then this ratio does not converge to 0.5.

4. Functions with Discontinuous Derivatives (Example 4) - It is also important to determine if

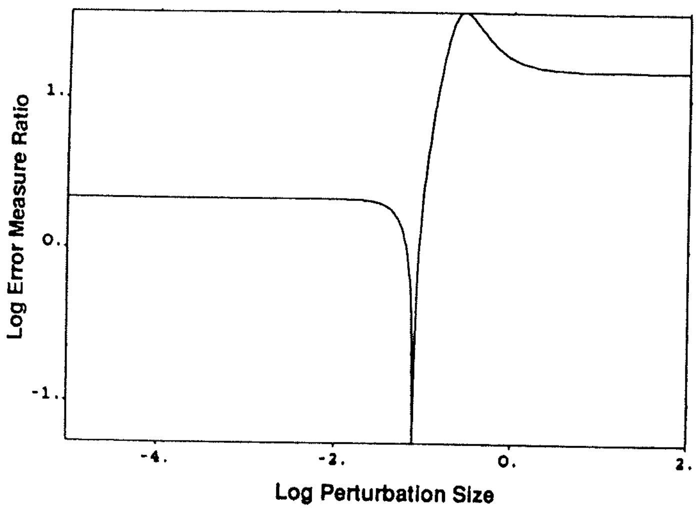

Figure 8: Ratio of the Error Estimates $e^{(\delta)}/\hat{e}^{(\delta)}$ in Example 2

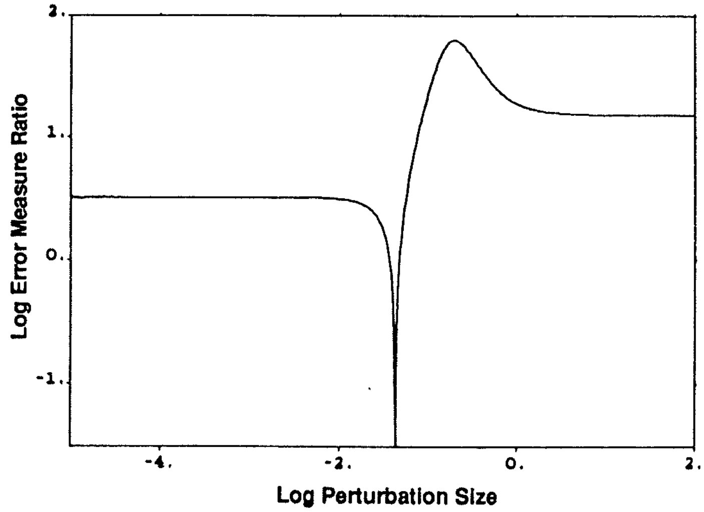

Figure 10: Ratio of the Error Estimates $e^{(\delta)}/\hat{e}^{(\delta)}$ in Example 3

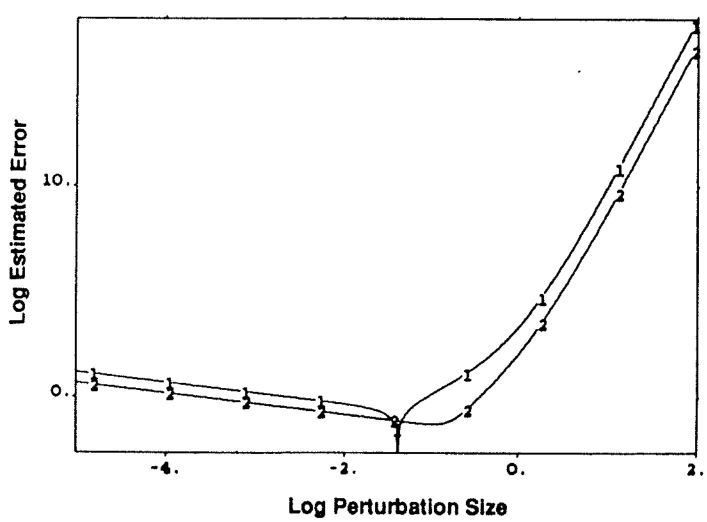

Figure 9: Error Estimates $e^{(\delta)}$ ('1') and $\hat{e}^{(\delta)}$ ('2') in Example 3

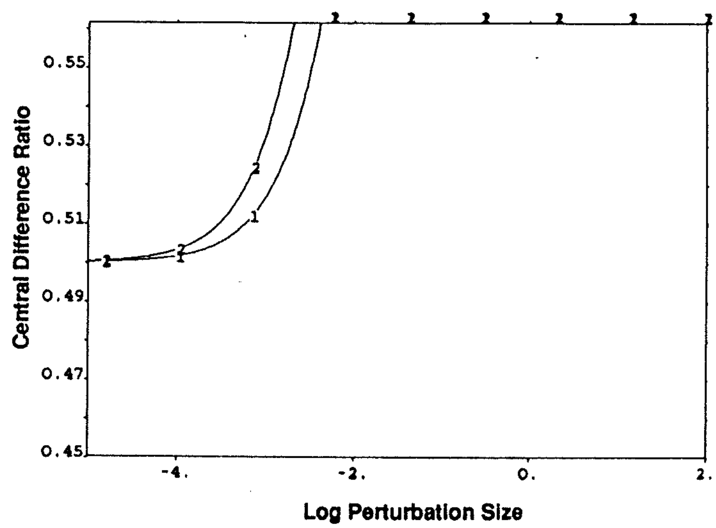

Figure 11: Central Difference Ratios $Df^{(\delta)}/Df^{(\delta/2)}$ ('1') and $Df^{(2\delta)}/Df^{(\delta)}$ ('2') in Example 2

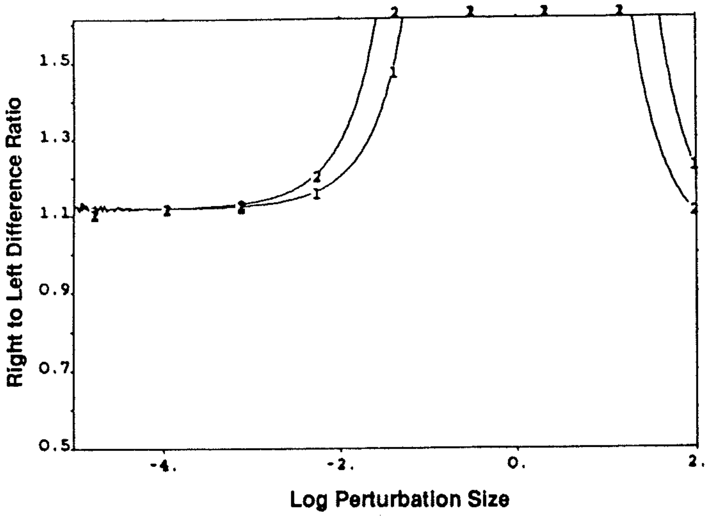

Figure 12: Df_R/Df_L with Perturbation δ ('1') and 2δ ('2') in Example 4

the derivative of a function is discontinuous at the point of differentiation. This can be done by considering the *right* and *left* derivative estimates, defined according to:

$$Df_R^{(\delta)} = \frac{f(x_0+\delta) - f(x_0)}{\delta} \quad (21)$$

and

$$Df_L^{(\delta)} = \frac{f(x_0) - f(x_0-\delta)}{\delta} \quad (22)$$

If the derivative of f is not continuous at x_0, then the right and left derivative estimates will not, in general, converge to the central difference estimate as the perturbation decreases. Example 4 with the ramp imposed at $x_0 = \sqrt[9]{2}$ exemplifies this type of nonlinear function. Figure 12 is a plot of the ratio of the right to left derivative estimate for this f as a function of δ. The fact that this ratio does not converge to unity is an indication that the derivative of f is not continuous.

The above observations provide the basis for testing and characterizing each nonlinearity in the model. This information is central to the development of the robust algorithm and logic in SIMNON+, as well as the rule-based system described elsewhere [6].

2.3.2. The SIMNON+ Routine OPTDELTA

The heuristic logic outlined below addresses the problem of robustly differentiating a scalar function of one variable, $f(x)$, at a point x_0. The extension to the general problem of linearizing vector functions of several variables, $\mathbf{f}(\mathbf{x},\mathbf{u})$, $\mathbf{h}(\mathbf{x},\mathbf{u})$ (Eqns. 1 and 2) about an operating point is straightforward.

1. Start with an initial $\delta_0 = 0.01$, unless the user specifies another value. Since the appropriate value of δ is dependent upon units and scaling, the user should be prepared to supply at least a reasonable default starting value, especially if 0.01 is blatantly inappropriate.

2. Determine if f is a constant, linear, or piecewise-linear function. In particular, we must distinguish between the linear and piecewise-linear case. This is done by comparing several derivative estimates over a wide range of perturbation sizes. To determine if f is linear evaluate and compare

$$Df_R^{(100\delta_0)} \quad Df_R^{(10\delta_0)} \quad Df_R^{(\delta_0)}$$

$$Df_L^{(100\delta_0)} \quad Df_L^{(10\delta_0)} \quad Df_L^{(\delta_0)}$$

   - If these six estimates are zero to within a tolerance based on machine precision, then declare the function to be constant (with respect to the variable under consideration). The derivative estimate is set to 0.0.
   - If these six estimates are the same to within a tolerance based on machine precision, then f has no curvature over a wide range and it is declared to be a linear function of x. The value of δ_0 is returned for use in calculating $\widehat{Df}^{(\delta)}$ (Eqn. 11).
   - If the right estimates are the same, and the left estimates are the same, but the right estimates differ from the left estimates by more than a user-supplied tolerance (percentage), then it is concluded that f is a piecewise linear function with a break-point at x_0. (This is a special case of having a discontinuous derivative - see Step 4 below.) The value of δ_0 is returned for use in calculating $Df_R^{(\delta_0)}$ and $Df_L^{(\delta_0)}$, and the 'diagnosis' is reported to the user, who can then decide which derivative estimate to employ (right, left, or average).

3. Check that a derivative exists at the point of differentiation, x_0. For example, f may be discontinuous, or its derivative may be infinite at x_0. In such instances round-off is almost nonexistent in the computation of derivative estimates, as mentioned above. Therefore, in checking for this condition, we consider error estimates for a very small perturbation size. Furthermore, it has been established (cf. Figs. 8, 10) that the ratio of the error estimates $e^{(\delta)}/\hat{e}^{(\delta)}$ is constant for small values of perturbation size. Thus, taking $\delta = \delta_0/100$, compare the ratios $e^{(\delta)}/\hat{e}^{(\delta)}$ and $e^{(2\delta)}/\hat{e}^{(2\delta)}$; if the difference between these ratios

is not within a given tolerance, then we conclude that f does not have an infinite derivative at x_0. If the ratios are the same to within the specified tolerance, then verify this result by taking $\delta = \delta_0/1000$ and repeating the above calculation. Now if the difference between all four of these ratios is within tolerance we check the ratio of the central difference with perturbation δ to that with perturbation $\delta/2$. If this value is within some neighborhood of 0.5, then we conclude that f is discontinuous at x_0 and report this finding to the user.

4. Compare the right and left derivative estimates at x_0, using perturbations of $\delta_0/100$ and $\delta_0/1000$. If the difference is not within some specified tolerance (percentage), then we conclude that the derivative of f is discontinuous at x_0, and inform the user, who would again decide which derivative estimate to accept (right, left, or average).

5. If, by the above tests, f is found to be nonlinear, continuous, and to have a finite continuous derivative, the following procedure is used to determine an optimum δ to estimate its derivative: First, take $\delta_1 = \delta_0/1000$ and assume that the derivative estimates are then dominated by round-off error. Then for each iteration increase δ_i by a factor of 2, until the ratio $\hat{e}^{(\delta)}/e^{(\delta)}$ is greater than 100 or until a maximum number of iterations has been exceeded. It was seen that when this ratio becomes greater than 100, δ is close to optimal (cf. Figs. 5, 6), so that value is used in calculating $\widehat{Df}^{(\delta)}$; if such a value of δ cannot be found, then the user is prompted to supply a new initial guess for the perturbation.

3. CONCLUSION

General algorithms and heuristics for robust numerical differentiation are required for using linear analysis and design methods on nonlinear systems. These have been obtained by developing and refining the linearization routine of SIMNON+ in parallel with the creation of rule-based systems for expert-aided modeling. As a result of this research, SIMNON+ now possesses a powerful routine to 'diagnose' the types of nonlinear relations in a system and then (when a function is found to be differentiable) to minimize the effects of truncation and round-off error incurred through the numerical differentiation process.

The error in estimating a derivative by computing a central difference is a function of the perturbation used. By measuring and comparing such errors for different perturbations, the SIMNON+ routine searches for the minimum of this function. In addition, a scheme for detecting discontinuities, discontinuous partial derivatives, and infinite partial derivatives has been implemented in SIMNON+. Tests have shown that the performance of the resulting SIMNON+ linearization routine is robust.

APPENDIX A. TRUNCATION ERROR IN DERIVATIVE ESTIMATION

It is shown here that a derivative estimate obtained by Richardson extropolation (Eqn. 11; [1,12]) eliminates truncation error through the fourth-order term of a nonlinear function expressible in a Taylor series about the point of differentiation. In addition, it is shown that the difference of two central-difference derivative estimates is a useful measure of the truncation error in the derivative estimates. Such a measure is indispensable for error control in the linearization process.

Consider the Taylor-series expansion for $f(x)$ about the point $x_0 = 0$ (with no loss in generality):

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + \cdots \quad (23)$$

Consider, then, the following two estimates of the derivative of $f(x)$ at $x_0 = 0$:

$$Df^{(\delta)} = \frac{f(\delta) - f(-\delta)}{2\delta} \quad (24)$$

$$Df^{(2\delta)} = \frac{f(2\delta) - f(-2\delta)}{4\delta} \quad (25)$$

Result: The derivative estimate of Eqn. 11 eliminates truncation error through the fourth-order term in the Taylor series of $f(x)$.

Proof: By direct substitution, a central-difference derivative estimate for $f(x)$ with perturbation size Δ (Eqn. 10) gives:

$$Df^{(\Delta)} = a_1 + a_3\Delta^2 + a_5\Delta^4 + a_7\Delta^6 + \cdots$$

Letting $\epsilon^{(\Delta)}$ denote the truncation error in the central difference above, it is evident that:

$$\epsilon^{(\Delta)} = a_3\Delta^2 + a_5\Delta^4 + a_7\Delta^6 + \cdots$$

We therefore have two instances of interest:

$$\epsilon^{(\delta)} = a_3\delta^2 + a_5\delta^4 + a_7\delta^6 + \cdots \quad (26)$$

$$\epsilon^{(2\delta)} = 4a_3\delta^2 + 16a_5\delta^4 + 64a_7\delta^6 + \cdots \quad (27)$$

From Eqns. 26 and 27 the truncation error in $\widehat{Df}^{(\delta)}$ (Eqn. 11) is:

$$\hat{\epsilon}^{(\delta)} = \frac{4}{3}\epsilon^{(\delta)} - \frac{1}{3}\epsilon^{(2\delta)} = -4a_5\delta^4 - 20a_7\delta^6 - \cdots \quad (28)$$

Eqn. 28 reveals that the truncation error in $\widehat{Df}^{(\delta)}$ is governed only by terms of order five or greater in the Taylor-series expansion of $f(x)$. *QED.*

It is now shown that the following error measure,

$$e^{(\delta)} = \frac{1}{3} \left| Df^{(\delta)} - Df^{(2\delta)} \right| \tag{29}$$

is a good indicator of the actual truncation error in $Df^{(\delta)}$ when the perturbation size is small. We see from Eqns. 26 and 27 that:

$$e^{(\delta)} = \frac{1}{3} \left| -3a_3\delta^2 - 15a_5\delta^4 - 63a_7\delta^6 - \cdots \right|$$

For small values of δ the lowest order term dominates, so $e^{(\delta)} \approx a_3\delta^2$, which (by Eqn. 26) is approximately the actual truncation error in $Df^{(\delta)}$.

[1] J. H. Taylor. Environment and methods for computer-aided control systems design for nonlinear plants. In *Proc. 2nd IFAC Symposium: CAD of Multivariable Technological Systems*, Purdue University, West Lafayette, IN, September 1982.

[2] H. A. Spang III. The federated computer-aided control design system. *IEEE Proceedings*, 72, December 1984.

[3] J. H. Taylor. Computer-aided control engineering environment for nonlinear systems analysis and design. In *Proc. 3rd IFAC Symposium on CAD in Control and Engineering Systems*, Lyngby, Denmark, August 1985.

[4] J. H. Taylor and D. K. Frederick. An expert system architecture for computer-aided control engineering. *IEEE Proceedings*, 72, December 1984.

[5] J. R. James, D. K. Frederick, and J. H. Taylor. On the application of expert systems programming techniques to the design of lead-lag precompensators. In *Proc. Control 85 Conference*, Cambridge, UK, July 1985.

[6] J. H. Taylor, J. R. James, and D. K. Frederick. Expert-aided control engineering environment for nonlinear systems. In *Proc. 10th IFAC World Congress*, Munich, FRG, July 1987.

[7] J. H. Taylor. Expert-aided environments for CAE of control systems. In *Proc. 4th IFAC Symposium on CAD of Control Systems*, Beijing, PR China, August 1988.

[8] H. Elmqvist. SIMNON—an interactive simulation program for nonlinear systems. In *Proceedings of Simulation '77*, Montreux, 1977.

[9] J. H. Taylor and P. D. McKeehen. A computer-aided control engineering environment for multidisciplinary expert-aided analysis and design (MEAD). In *Proc. National Aerospace Electronics Conference (NAECON)*, Dayton, OH, May 1989.

[10] J. H. Taylor and K. L. Strobel. Nonlinear compensator synthesis via sinusoidal-input describing functions. In *Proc. 1985 American Control Conference*, Boston, MA, June 1985.

[11] J. H. Taylor and J. R. O'Donnell, Jr. Synthesis of nonlinear controllers with rate feedback via SIDF methods. In *Proc. 1990 American Control Conference*, San Diego, CA, May 1990.

[12] G. Dahlqvist and A. Bjorck. *Numerical Methods.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1974.